AD-A086 329 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2

A STUDY OF LANDSAT MULTI-SPECTRAL SCANNER DATA TAPES.(U)

FEB 80 A H BENNY

UNCLASSIFIED RAE-TR-80018 DRIC-BR-73767 NL

1 OF 1

AD

A086 329

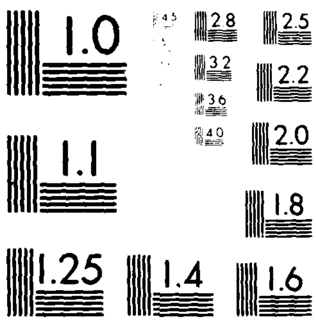
END

DATE

FILMED

8-80

DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

ROYAL AIRCRAFT ESTABLISHMENT

*

Technical Report 80018

February 1980

A STUDY OF LANDSAT MULTI-SPECTRAL SCANNER DATA TAPES

by

A.H. Benny

*

Procurement Executive, Ministry of Defence
Farnborough, Hants

UDC 629.195 : 339.5 : 531.7.084.2 : 535.61 : 681.3.053

(18) DRIC

(19) BR-73767

ROYAL AIRCRAFT ESTABLISHMENT

(9) Technical ~~Report~~ 80018 (14) RAE-TR-80018

Received for printing 11 February 1980

(11) Feb 80 (12) 38

(6) A STUDY OF LANDSAT MULTI-SPECTRAL SCANNER DATA TAPES

by

(10) A. H. Benny

SUMMARY

A detailed study is described of a number of computer compatible magnetic tapes containing Landsat multi-spectral scanner (MSS) data. This study reveals the differences between raw and corrected data tapes and indicates how the conversion is performed. It also shows the differences between the existing and a new form of corrected data tape.

From the knowledge gained, a computer program has been written, to enable raw data, desirable for certain computer analyses, to be obtained from the existing stocks of corrected data tapes.

Departmental Reference: Space 574

310450

LIST OF CONTENTS

1 INTRODUCTION

Amongst the many satellites at present orbiting the Earth is a series known as Landsat. These vehicles contain a number of sensors which make observations of the surface of the Earth, the information obtained being broadcast by the satellites and collected at ground receiving stations. The type of sensor of interest in the present study is the multi-spectral scanner (MSS), which is described in Ref 1.

Normally, the received data is recorded on high density magnetic tapes which are not computer-compatible, that is they are not capable of being read directly into a conventional computer installation. It is therefore common for ground receiving stations to have the special equipment necessary to convert high density tapes to the more widely usable computer compatible tapes (CCTs).

The conversion from high density tape to CCT may be done with little or no alteration to the data content, and such a CCT may be thought of as containing 'raw' or unprocessed data. Since most users require corrected data tapes, it is common practice for the high density tapes to be converted directly into 'corrected' CCTs, without intermediate storage in 'raw' form.

Receiving stations may therefore offer to supply Landsat MSS data users with either raw or corrected data tapes. Until recently, the demand has been solely for corrected tapes. For example, Space Department, RAE has acquired a considerable number (about 200) of corrected CCTs but had no raw CCTs at the time that this work was started.

During the course of computer analysis of Landsat MSS data it has become apparent that corrected data suffers from certain adverse features, resulting from the correction process. It was therefore considered desirable to obtain raw data, and in view of the considerable amount of corrected data held at RAE it seemed worthwhile to attempt to devise a method (ie a computer program) to restore raw data from corrected tapes. This Report describes a study which has been made to determine the data-structure of raw and corrected tapes and the relationship between them, with the aim of converting corrected data back to the raw form.

1.1 Cautionary note

Many scientific investigations are concerned with natural objects or phenomena, whereas this study is concerned with man-made objects (CCTs). There is a possibility that a purely descriptive observation may be interpreted as

implying adverse criticism. The author wishes to make it clear that this Report is not intended in any way as a criticism of the tapes under study.

2 DESCRIPTION OF COMPUTER COMPATIBLE TAPES

The 'corrected' computer compatible tapes are described in some detail in Ref 2, dated May 1976 and redefined in Ref 3 dated October 1979, which became available during the course of the work described. These publications give considerable detail about the tapes, though they leave a number of small but important points unresolved. No description of 'raw' data is given, and insufficient details of the conversion from raw to corrected data are provided to allow the process to be reversed.

A comparison of Refs 2 and 3 shows that significant differences exist between the relevant formats, so this study will refer to 'old' and 'new' corrected tapes, the former being produced before the summer of 1979 and the latter after that time. On the basis of information received by private communication, the changeover date has been selected as 1 August 1979.

When the present study began, it was not known that a format change was about to occur. It is perhaps fortunate that the change occurred at this time, as this tudy was then extended to cover the new format, and has been able to be more complete in itself and in the output products (ie computer programs) which have resulted from it.

2.1 General structure of the CCTs

The general structure of all CCTs, both raw, and old and new corrected, is similar, and is illustrated in Table 1. The magnetic tapes contain a sequence of 'records', as shown, starting with two header records, separated by a tape marker. The next six records refer to transformations and calibrations relevant to the scene, and are followed by a tape marker. There are 9144 data records, referring to the four spectral bands of data for the 2286 lines or scans of the scene.

Refs 2 and 3 describe in some detail the nature and contents of these records. This Report is concerned only with details of the records not clarified by these references, and with differences between the various raw and corrected tapes. If it is intended to study section 3 in detail, it is advisable that either Ref 2 or 3 be available for concurrent use.

2.2 The data tapes examined

Space Department, RAE had an extensive collection of corrected CCTs all of the 'old' type, when this study started. Two more tapes were obtained especially for this study, these being a raw and a corrected tape (nominally of the same scene). As the study proceeded, it became apparent that the corrected tape was of a different and new type. Since then, several other tapes have been examined, the details being shown in Table 2. All of these tapes were processed at Fucino in Italy, over a period of some 21 months. The tapes referred to as D and E contain data from Landsat 1, the others from Landsat 2. The scene numbers refer to the satellite path and row[1]. Tapes A and B are the same scene, a portion of Sweden, and both derive from the same satellite pass. The other scenes are of the south-west of England, including the Severn estuary.

3 DETAILED INVESTIGATION OF THE TAPES

The computer compatible magnetic tapes were examined in detail, on a Prime 400 computer system including a 1600 bpi 9-track magnetic tape unit.

A computer program was available which allowed the tapes to be studied in close detail. However, due to the very large quantity of data, it was not possible merely to print it all out and study the listing, as this would have run to many thousands of closely packed pages for each tape. Instead, several computer programs were written, to explore various facets of the information required. Some of these programs are mentioned in the course of this and the next section.

This section is not to be taken as a complete description of the tape formats, for which attention should be turned to Ref 2 or 3. Mention will only be made here of detail not noted in those references.

3.1 JSC header record

The JSC or Johnson Space Centre Header Record contains 3060 bytes, most of which are unused. Only two items have been found in the JSC header on all tapes, but some tapes have a few additional items.

The items specified in Ref 3 are listed in Table 3. All of the tapes studied contain items 1 and 2. It is of interest that the values provided on tapes A and B, which are nominally the same scene, differ slightly. This may be because the scene is in fact slightly displaced, between tapes A and B, or it may reflect a more accurate value in the case of the corrected tape B.

The old corrected tapes examined all contain item 3 and the new corrected tapes also contain item 4.

3.2 Landsat header

Whilst the JSC header always uses EBCDIC characters, the Landsat header and subsequent records may use either EBCDIC or ASCII. The character code is indicated by one of the process flag bits in the Landsat header. All of the tapes examined have been found to use EBCDIC characters.

The contents of the Landsat header are described in Refs 2 and 3. The differences noted between the headers on the raw tape (A) and corrected tape (B) for a nominally identical scene are:

		Value	
Item	Contents	Raw	Corrected
1	Day number since launch	899	1899
2	Frame identification	09012	2208018049
3	Centre latitude N	5990	5984
4	Centre longitude E	-1799	1768
5	Cycle number	50	49
6	Date master tape generated	5 April 79	16 Sept 79
7	Date copy tape produced	5 April 79	16 Sept 79
8	Process flags	0000000	1111010

Comments may be made, on the differences between the two.

Item 1 Raw data omits leading '1': this is the true value, whereas the corrected value is in error.

Item 2 The frame identification on corrected tapes includes the mission (Landsat 2) track (208) frame (018) and cycle (049).

On the raw tape it may refer to hours (9) minutes (01) and tens of seconds (2) at which the centre point was imaged, ie 9:01:20 on the day specified.

Item 3 and 4 The small differences here probably represent the slightly different area imaged (see section 4.1). The differences are of the required order of magnitude, for a satellite near to the northerly apex of its orbit, assuming that the values represent degrees (two figures) and hundredths of degrees (two figures). The specifications state that the second pair of figures represent minutes of arc, but this clearly cannot be the case, as the values exceed the maximum possible value of 59 minutes.

018

The minus sign in the raw data for item 4 is not understood: it is clearly in error.

Item 5 It is not known why these values differ. The actual cycle number is 49, which means that the raw data tape is incorrect.

Items 6 and 7 These are the dates at which the CCTs were prepared, and of course need not be the same.

Item 8 Raw data is 'unprocessed' and would be expected to have process flags all zero. The code for the processed tape implies: (left to right).

Bit 1: No meaning to user
Bit 2: 256 radiometric levels
Bit 3: Geometric correction applied
Bit 4: No meaning to user
Bit 5: Statistical radiometric correction applied (as distinct from step wedge calibration).
Bit 6: No meaning to user
Bit 7: EBCDIC character code (as distinct from ASCII)

It is noteworthy that the other corrected tapes have the same process flags as tape B.

3.3 Transformation record

The transformation record contains up to 36 values, but in the tapes studied so far only seven items have non-zero values. The values on the raw and corrected tapes are:

Item	Meaning	Raw	Corrected
1	UTM zone number	33	33
2	UTM Northing (metres)	6643843.5	6636431.27
3	UTM Easting (metres)	667227.25	650377.3597
4	Frame orientation	0.4261635542	0.275119713
5	Pseudo altitude of satellite (km)	249.7637482	248.7237
7	X scale factor	3.703999996	3.703999996
8	Y scale factor	925.124939	3.703999996

Items 2, 3 and 4 would be expected to differ somewhat, because of the difference between the scene framing, mentioned earlier. Item 5 may, or may not, depend on the same reason.

Items 7 and 8 refer to the 70 mm film images which may be produced as an alternative to CCTs, so probably have no relevance to the computer processing done at RAE.

3.4 Radiometric look-up table records

There are five look-up table records, corresponding to the spectral bands 4 to 8 inclusive. This study is concerned only with bands 4 to 7, so band 8 is not considered here.

The look-up tables show the conversion which has been used to transform 6-bit values (0 to 63) from the satellite sensors to 8-bit values (0 to 255) on the corrected tapes. The conversion takes account of the known behaviour of the sensors (ie bands 4 to 6 are quasi-logarithmic, whereas band 7 is linear) and also the differences in sensitivity between each of the six detectors associated with each spectral band.

For any scene, a set of look-up tables may be prepared. The 6-bit data on the high-density tapes is either used unaltered (for raw tapes), or transformed to 8-bit data by means of the look-up tables (for corrected tapes).

The contents of the look-up tables have been examined on the various types of tape, and the differences are described below.

3.4.1 Raw tapes

For raw data tape A all of the look-up tables contain only zero values. (This is to be expected since raw data does not need such tables.)

3.4.2 Corrected tapes

For corrected tapes, complete look-up tables are provided, as expected. Ref 2 explains that the tables are derived either from calibration data from the satellite (calibration wedge), or from a statistical analysis of the scene itself. Bit 5 of the process flag word in the Landsat header indicates that the latter has been done for all corrected tapes studied.

It is of interest to note that the look-up tables contain a few instances of two adjacent values being the same. Thus it is not always possible to perform the inverse mapping from corrected to raw values without an occasional ambiguity occurring. This does not occur on all images, and it is possible to restore data to its raw form in many cases, at least as far as the radiometric values are concerned.

3.5 Multi-spectral scanner data records

A scene is defined as 2286 lines. Each line consists of four spectral bands, each of which is contained in one magnetic tape record. There should therefore, be 2286 x 4 = 9144 data records on the tape, followed by a file marker. At 1600 bpi this almost fills a 2400 foot tape.

The data structure may, therefore, be considered as consisting of sets of four records (bands 4, 5, 6 and 7) for each line of the scene. Of each set of four, the first record differs from the other three, in the manner noted below.

Each record consists of 3780 bytes. Bytes 1 and 2 form a 16-bit integer, having the values 1, 2, 3, 4 for each of the records of a set of four. Thus, byte 1 of the record is always zero, and byte 2 of each band 4 record, for example, would have the value 1.

For band 4 data, bytes 3-180 contain 'ancillary data' (see below) and bytes 181-3780 contain video data. For the other three spectral bands, the video data is contained in bytes 3-3602 and the ancillary data is in bytes 3603-3780, ie after the video data. The ancillary data for bands 5, 6 and 7, where it exists is usually the same as that for band 4.

3.5.1 Ancillary data

The raw data tape A contains ancillary data only in the band 4 record; in the other three bands the words are all zeros.

The interpretation of the ancillary data is described in Refs 2 and 3, but the meaning of several items is not entirely clear. For instance:

'End of video flag count' - on the raw data tape this has a value which may be interpreted as -19700.

'Minor frame sync loss count' - zero in all cases examined.

'Uncorrected line length' - this seems to be approximately but not exactly equal to the number of pixels in the line of raw data which corresponds to the corrected line. On the raw data tape the value in the ancillary block is zero.

The meaning of 'start pixel number' and 'end pixel number' seems to vary from one type of tape to another. This is described in more detail in section 3.5.2.

Summarizing the ancillary data therefore:

(a) Raw tapes only contain ancillary data for band 4; for other bands it is all zero. Even in the band 4 data, only a few of the values are present

namely: time and day/time, EOV flag, scan number, start and end pixel numbers, and sensor number.

(b) Old corrected tapes contain ancillary blocks for all bands, but those for bands 5, 6 and 7 are less complete, eg they lack scan line numbers, start and end pixel numbers, time and day/time. For band 4 the EOV flag is set to -19700 (as with the raw data), and the XSCAN is set to zero. The 'one fills' are also set to zero but most of the other ancillary data is provided with the exception of the uncorrected line length.

Tape D, which was processed some months before tape C, has less data in its ancillary blocks. In particular, the time and day/time are zero, even for band 4. On this tape the EOV flag is -27636.

It seems possible, therefore, that the corrected data tapes have gone through a number of versions, each more complete than the previous one.

(c) The new corrected tape has almost complete ancillary blocks for all bands. The scene does not start with Sensor No.1. The XSCAN values represent the distance along the swath in metres; 0, 80, 160 etc, for rows 1, 2, 3 etc.

3.5.2 Pixel start and end numbers

Each MSS data record has 3600 bytes available for radiometric data. As mentioned, the data is stored in bytes 181 to 3780 for band 4, and bytes 3 to 3602 for bands 5, 6 and 7.

Normally, considerably less than 3600 radiometric values are provided: the actual number is quoted as being between 3000 and 3450 bytes, and in practice is about 3244 on the raw data tape. The actual number of values depends upon the interval between the start-of-scan and end-of-scan signals in the satellite, and seems to vary very little over a considerable period of time.

All values are stored as binary, and bytes which do not contain radiometric values are set to zero, such values occurring both before and after the radiometric values. Since zero is a valid radiometric value it is not always possible to distinguish the start and end of data, by inspecting the bytes. It is therefore necessary for the system to specify where radiometric values start and end.

All of the tapes contain pixel start and end numbers in the ancillary block for band 4 of every row of the scene. In some cases (eg tape B) the ancillary blocks for bands 5, 6 and 7 also include start and end numbers, and these are always the same as those given for the corresponding band 4. However, the interpretation is different, in some cases for every band, as described next, where:

SP = start pixel number (bytes)

FP = finish or end pixel number (bytes)

(a) Raw data tape A

Band	Start at	End at
4	SP	FP + 178 + 4
5	SP-178	FP + 2
6	SP-178	FP
7	SP-178	(FP-2)

The end for band 7 is deduced. Since the radiometric values at the end of the line were zero it was not possible to determine the end with certainty.

For the first row of the scene SP = 411 and FP = 3473 so:

Band	Start	End	No. of pixels
4	411	3655	3245
5	233	3475	3243
6	233	3473	3241
7	233	(3471)	(3239)

The values of SP and FP were examined for the whole of the tape. SP remained at the value 411 throughout the entire scene, whilst FP alternated between the values 3473 and 3474, the change occurring after irregular numbers of lines, eg at lines 19, 25, 79, 85, 91 etc.

(b) New corrected data tape B

Band	Start at	End at
4	SP	FP
5	SP-178-2	FP-178-2
6	SP-178-4	FP-178-4
7	SP-178-6	FP-178-6

For the first row of this tape SP = 454, FP = 3753, hence:

Band	Start	End	No. of pixels
4	454	3753	3300
5	274	3573	3300
6	272	3571	3300
7	270	3569	3300

018

This tape gives the 'uncorrected line length' as 3240, for band 4 row 1. This does not refer to the number of pixels in the line (which is seen, above, to be 3300) but corresponds approximately to the number of pixels in the line of raw data (see (a) above).

The values of SP and FP were found to decrease progressively throughout this tape. For example, SP = 454 and FP = 3753 (for all four ancillary blocks) for the first row of the scene. Subsequently, SP and FP both decreased by one, at about every 20th row of the scene, ending the scene with values SP = 348 and FP = 3647. In every case the difference FP-SP was the same, 3299, ie a constant line length of 3300 pixels.

The decreasing SP and FP values imply that the scene as a whole is a parallelogram rather than a rectangle, this being a result of geometric correction for Earth rotation below the moving satellite. It is normal practice in RAE to extract the maximum possible rectangle from this parallelogram.

(c) Old corrected data tape C

Band	Start at	End at
4	SP	FP + 178 + 8
5	SP-178	FP + 6
6	SP-178	FP + 4
7	SP-178	FP + 2

For the first row, SP = 321, FP = 3415 so:

Band	Start	End	No. of pixels
4	321	3601	3281
5	143	3421	3279
6	143	3419	3277
7	143	3417	3275

In the older corrected tape, the uncorrected line length is not given. The number of pixels for each band reflects the varying number in the raw data. However, in the new corrected tape (B) the number of pixels has been made constant (3300) for each band.

It is thus apparent that there is an older (C) and newer (B) type of corrected data tape, which need different interpretation, at least concerning the start and end pixels.

4 ADDED PIXELS

It has been shown that the raw data scene has about 3240 pixels per row, whereas the corrected scene has 3300. The reason for this is that extra pixels have been inserted, in a systematic manner, to 'correct' the geometry of the scene.

There are two major distortions in an MSS scene which affect the distribution of pixels along a line. Firstly, the oscillating mirror in the Landsat satellite does not move at a constant speed, and this gives rise to 'non-linear mirror velocity distortion'. Secondly, because of the 5.9 degree field of view on either side of the vertical, there is the so-called 'panoramic distortion'. These two distortions have a cumulative effect, which is to 'stretch out' the edges of the scene, in relation to the central regions.

To compensate for these distortions, the corrected tapes have pixels added to them. Pixels are added only infrequently at the edges of the scene (ie near the ends of the lines) but increasingly more are added near the centre, for example at a rate of one extra pixel per 28 pixels. This effectively 'stretches' the central regions, so that the whole scene is linearly compensated.

4.1 Comparison of raw and corrected tapes

Information about the added pixels can be obtained by comparison of raw and corrected tapes. Each line of a corrected scene should correspond to the same line of the raw scene, but with the radiometric values transformed by means of the look-up tables and with added pixels.

When the two tapes were studied it became apparent that the scenes, although nominally similar, were not identical, and it was eventually determined that line 1 of the corrected tape corresponded to line 262 of the raw data tape. With this knowledge it was then possible to list the pixel values for the corresponding lines.

The values of the two lines would not be expected to correspond directly, because the raw values are converted, by means of a look-up table, to corrected values. When this conversion was taken into account, the two lines could be compared. It then became apparent that extra pixels were in fact being added, by duplication of preceding values, at intervals throughout the line.

Following this discovery, a program was written to detect all pixels which had the same value as the preceding one. Clearly this would detect both added

pixels and other, genuine duplicates. However, by repeating the process for several lines, it was possible to detect the locations of the added pixels.

When this program was run on 50 lines of the corrected tape B (10 was insufficient) it showed very clearly where the pixels were added. At the ends of the line, one pixel was added for about every 100 existing ones, and at the centre one pixel per 30 was added, (ie 30 pixels became 31), with 79 pixels added in all.

When more extended tests were run, it became apparent that the sequence of added pixels was changed at intervals throughout the scene. Further examination indicated that this change occurred after every 82 lines. It is not known whether this interval is common to all new corrected tapes, or just to this particular tape; verification would be a lengthy process.

4.2 Study of older corrected tape

An older corrected data tape (C) was then studied.

When the program described above was run, no 'added pixels' were found. Further studies indicated that pixels were being added by averaging, rather than by duplication of existing ones, ie each added pixel was assigned a value which was the average of the pixels on either side of it.

It seemed that the best approach would be to inspect the look-up table appropriate to the band and sensor, and then examine the line for radiometric values not in the look-up table. This would not detect all added pixels, as sometimes the average would be an existing value, but if repeated on several lines it should reveal added pixels only.

Another program was prepared to perform this task, and this successfully detected the added pixels. As with the tape B, tape C shows that pixels are added more closely in the centre of the line, and less frequently at the ends, but the added pixels are of course averaged rather than duplicated values.

Adding 'averaged' pixel values is not acceptable when computer interpretation is to be done. For example (and for simplicity the example will be confined to one band, - although it is equally applicable to all 4 bands) consider two types of feature on a scene, feature 1 having radiometric values of say 10-20 and feature 2 having values of 90-100. If a pixel is inserted between two such features, an average value of say 50-60 will be provided, which is unlike either feature 1 or feature 2, but may be consistent with some other type of feature. Thus adding averaged pixel values is a form of data corruption.

4.3 Comments on added pixels

Geometric correction of any kind, on a raster-scan image, must inevitably give rise to some amount of data degradation, so for some purposes, the raw, and hence distorted, scene is preferable. For many users, however, including those performing a computer analysis, geometric correction may have some advantages. This study has shown that the corrected tapes use added pixels to obtain a geometrically corrected scene, and that the 'old' method was to add averaged values, whilst the 'new' method is to insert duplicate values. Both methods have disadvantages, but it seems likely that the new method is less undesirable.

This study has shown that it is in practice possible to detect the locations of the added pixels with both old and new tapes. In the case of the new tape the change after every 82 lines makes this detection very laborious. It is, however, practical to use a computer to detect and convert the added averaged pixels of the old tapes to added duplicate pixels, and such a conversion may often by advantageous, for image analysis purposes.

5 PROGRAMS TO PROCESS TAPES

The purpose of this work was to obtain information about Landsat MSS data tapes, so that 'raw' information could be extracted from the extensive stock of corrected tapes held at RAE. During the course of the work, the 'new' corrected format became the standard, and consequently demanded a new program to read it correctly.

Consequent upon this study, therefore, two new programs have been prepared and these are described in sections 5.2 and 5.3.

5.1 Disc image files

When it is intended to perform any computer analysis of a Landsat MSS scene, it is convenient to read the CCT and form a set of image files, held in disc store. This is done chiefly for speed of access, as any portion of a disc file can be accessed in much less than one second, as distinct from a magnetic tape where access can take several minutes.

It is convenient to form four equal sized disc files, of bands 4 to 7 inclusive. Each file consists of a 'header' followed by the video data held as a raster scan, stored in sequence, and all files are 'in register', that is to say they correspond to exactly the same portion of the Earth's surface. Failure to achieve registration will of course falsify any inter-band analysis.

Due to the mode of operation of the Landsat Multi-spectral Scanner, the pixels in any line are displaced by two from band to band (see Ref 1). The specifications mentioned in Refs 2 and 3 do not make clear which pixels are in register. During the development of the programs described below, it has been necessary to examine the disc images, to check for registration, and to modify the program accordingly. Registration is most easily checked by selecting some image feature which is conspicuous in all four bands, and ensuring that its location is identical on all four image files.

5.2 Program to convert corrected to raw data

Since 'correction' includes both the conversion of values, using look-up tables, and the addition of pixels for geometric correction, two kinds of 'uncorrection' are possible: these being of course the removal of added pixels and the restoration of the original radiometric values. It would be possible to write a program to perform one or both of these operations for the 'old' format, and another program to do the same for the 'new' format.

It was finally decided to prepare a program which could operate on either old or new tapes. This program examines the date of production of the tape, which is included in the Landsat header, and decides whether it is old or new according to whether the production date is before or after 1 August 1979. Having decided this, the appropriate conversion is selected.

For many purposes it is desirable to retain the geometric correction, and for new tapes the added (duplicate) pixels are acceptable. However the added (averaged) pixels of the old tapes are not considered acceptable, and thus these are replaced by duplicate values. In the case of both old and new tapes the radiometric values are then restored to their raw values by operation of the look-up tables in reverse.

The complete program is named IM.UNCOR, consistently with the naming of the other image programs in Space Department. The listing of this program is attached in Appendix A.

5.3 Program to read old and new corrected tapes

Before this work began, there was a program, named IM.FUC.MSS, which was used to read the old (then standard) MSS tapes produced in Fucino and to form image files from the data.

A new program has now been prepared, named IM.FUC.MSS.2 which incorporates both the facilities of the earlier program, and in addition the ability to

process the new tapes. As with IM.UNCOR, the program itself decides whether the tape is an old or new one, and proceeds accordingly.

The opportunity has been taken to incorporate a number of features in this program which make it considerably more convenient to use than the original one, namely:

(a) The program determines whether the character code used is EBCDIC or ASCII, instead of asking the user. The information is obtained from one of the process flag bits in the Landsat header record.

(b) The program only asks for one disc file name, and suffixes it with 4, 5, 6 or 7 according to the band in question. Likewise it only asks for one 'text' for the file header, and attaches a suffix. Previously the user had to provide four file names each with text.

(c) The file names and texts were previously requested after the first pass through the tape had been done, ie about 30 minutes after the start of the program. Now the questions occur at the start of the run, so that no waiting is involved.

(d) The user can specify which bands he wishes to use, whereas previously he had to have all four. This is advantageous if the space available on disc is limited.

(e) The user can specify what portion of the image he wishes to use - eg lines 1000 to 2000, whereas before the whole image was input.

A listing of this program is given in Appendix B.

6 CONCLUSIONS

A detailed study has been made of a raw Landsat MSS CCT and several corrected tapes, both old and new. This study reveals the differences between the various types of tape, and indicates how the conversions from raw to corrected form have been made.

From the knowledge gained of the detailed structure of these tapes, it has been possible to determine how to convert corrected tapes (of which RAE Space Department has an extensive stock) back to raw data form. A program has been written to do this for both old and new corrected tapes. The resulting raw data is more suitable than corrected data for certain computer analyses.

As a result of this work, a new version of the tape to disc file input program has also been prepared. This has several operating advantages over the

existing program in ease and versatility of use, in addition to being capable of processing both old and new tapes, whereas the existing version could only handle the old tapes.

Appendix A

LISTING OF PROGRAM IM.UNCOR


```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE
C FOR OLD TAPES IT CONVERTS INTERPOLATED ADDED PIXELS TO DUPLICATES
C THEN (OLD OR NEW) IT CONVERTS PIX VALUES BACK, USING INVERSE LOOK-UP TABLES.
C THE OUTPUT IS ONE TO FOUR IMAGE FILES
      PARAMETER M=3700  /*NO. OF BYTES PER IMAGE BLOCK
      LOGICAL EOT,EOF,ERR,Q,ASCII,NEW,FLAG,PRINTH
      INTEGER*4 DNUM
      DIMENSION IBUFF(M),ISTORE(M),NAME(16),ITEXT(40)
      DIMENSION NBN(4),NOS(4,6,64),NBAK(4,6,256)
$INSERT SYSCOM>ASKEYS
      DATA SUFFIX /4H4567/
      DATA NLINES/50/  /*NO OF LINES TO CHECK FOR ADDED PIX
      DATA IS,IF/54,55/  /*USED FOR CALC OF ISP,IFP
C
C
C
      WRITE(1,1000)
 1000 FORMAT('IM.UNCOR   10-JAN-80')
C
   20 Q=RNUMSA('INPUT TAPE UNIT NUMBER',22,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 20
      IMUNIT=INTS(DNUM)
      IF (IMUNIT.LT.0 .OR. IMUNIT.GT.3) GOTO 20
      CALL REW(IMUNIT)
C
      DO 30 NBAND=1,4
   30 NBN(NBAND)=0  /*NO BAND IN USE..
      Q=YSNOSA('ALL BANDS',9,A$NDEF)
      IF (.NOT.Q) GOTO 33
      DO 32 NBAND=1,4
   32 NBN(NBAND)=1
      GOTO 40
   33 Q=RNUMSA('HOW MANY BANDS',14,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 33
      NBANDS=INTS(DNUM)
      IF (NBANDS.LT.1 .OR.NBANDS.GT.3) GOTO 33
      DO 39 J=1,NBANDS
   35 Q=RNUMSA('BAND NO',7,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 35
      JB=INTS(DNUM)
      IF (JB.GE.4 .AND. JB.LE.7) GOTO 36
   37 WRITE(1,1002)
 1002 FORMAT('REPEAT THAT NUMBER')
      GOTO 35
   36 JC=JB-3
      IF (NBN(JC).EQ.1) GOTO 37
      NBN(JC)=1
   39 CONTINUE
C
   40 LSTART=1  /*START FOR ENTIRE IMAGE
      NLINE=2286  /*NO. OF LINES IN ENTIRE IMAGE
      LPI=NLINE  /*END LINE FOR ENTIRE IMAGE
      Q=YSNOSA('ENTIRE IMAGE',12,A$NDEF)
      IF (Q) GOTO 46
   42 Q=RNUMSA('STARTING LINE',13,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 42
      LSTART=INTS(DNUM)
      IF (LSTART.LT.1 .OR.LSTART.GT.2286) GOTO 42
   44 Q=RNUMSA('HOW MANY LINES',14,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 44
```

```
C PROGRAM IN.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


      NLINE=INTS(DNUM)
      IF (NLINE.LT.0 .OR. NLINE.GT.2286) GOTO 44
      LPI=LSTART+NLINE-1  /*LAST LINE OF OUTPUT IMAGE
      IF (LPI.GT.2286) GOTO 42
C
C READ TEXT AND FILENAME FROM OPERATOR
   46 WRITE(1,1004)
 1004 FORMAT('ENTER TEXT MESSAGE FOR IMAGE')
      READ(1,1005)ITEXT
 1005 FORMAT(40A2)
   47 Q=RNAMSA('OUTPUT FILE NAME',16,A$FUPP,NAME,32)
      IF (.NOT.Q) GOTO 47
C
C READ THROUGH HEADERS, WITH OPTIONAL PRINTOUT
      PRINTH=YSNO$A('PRINT HEADERS',13,A$NDEF)
      CALL MTRE(INUNIT,IBUFF,1530,EOT,EOF,ERR,NW)  /*READ BLOCK 1
      IF (PRINTH) CALL JSCPT(IBUFF)  /*WRITE JSC HDR TO VDU
      CALL MTRE(INUNIT,IBUFF,1530,EOT,EOF,ERR,NW) /*READ EOF
      IF (.NOT.EOF) WRITE(1,1000)
 1000 FORMAT('NOT EOF FOLLOWING JSC HEADER')
      CALL MTRE(INUNIT,IBUFF,720,EOT,EOF,ERR,NW) /*LANDSAT HDR
      CALL LANPT(IBUFF,ASCII,NEW,PRINTH)  /*LANDSAT HEADER
      IF (NEW) WRITE(1,1001)
 1001 FORMAT('THIS IS A "NEW" TAPE - PRODUCED AFTER JULY 79')
      CALL MTRE(INUNIT,IBUFF,360,EOT,EOF,ERR,NW) /*TRANSFORMATION RECORD
      IF (PRINTH) CALL TRANPT(IBUFF,ASCII)  /*WRITE TRANSFORMATION RECORD
      DO 49 J=1,5  /*DATA RECORDS 4 TO 8
      CALL MTRE(INUNIT,IBUFF,810,EOT,EOF,ERR,NW)  /*LOOK-UP RECORDS
      IF (J.EQ.5 .OR. NBN(J).NE.1) GOTO 49
      JB=J+3
      IF (PRINTH) CALL LOOKPT(IBUFF,JB,ASCII)  /*WRITE A TABLE
      IF (.NOT.PRINTH .AND. .NOT.ASCII) CALL EBCASC(IBUFF(1),810,0)
      DECODE(1536,1111,IBUFF)ISTORE   /*TEMPORARILY INTO ISTORE
 1111 FORMAT(384(I4))
      DO 48 K=1,6
      KA=(K-1)*64
      DO 482 L=1,256
  482 NBAK(J,K,L)=-1  /*SET TO -1 IF NO VALUE EXISTS
      INO=0
      DO 484 L=1,64
      IN=ISTORE(KA+L)
      NOS(J,K,L)=IN
      IF (INO.LT.255) NBAK(J,K,IN+1)=L-1  /*LOOK-UP VALUES FOR UN-CORRECTING
      INO=IN
  484 CONTINUE
   48 CONTINUE
   49 CONTINUE
      CALL MTRE(INUNIT,IBUFF,1530,EOT,EOF,ERR,NW) /*READ EOF
      IF (.NOT.EOF) WRITE(1,1003)
 1003 FORMAT('NOT EOF AFTER LOOK-UP TABLES')
C
      NWB=1890
      NWB1=NWB+1
      ISENS=0  /*START AT SENSOR 1 FOR OLD TAPES
      LINET=0  /*COUNT TEST LINES
      DO 52 I=1,N
   52 ISTORE(I)=0
      DO 100 LINE=1,LPI
      ISENS=ISENS+1
      IF (ISENS.GT.6) ISENS=1
```

```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


      DO 80 IBAND=1,4
      CALL MTRE(IMUNIT,IBUFF,NWB,EOT,EOF,ERR,NW)
      IF (EOF) GOTO 990
      IF (IBAND.NE.1) GOTO 60
C CHECK ON SENSOR NUMBER
      ISS=RS(IBUFF(60),8)  /*BYTE 117 IN ANCIL BLOCK
      IF (.NOT.NEW .OR. LINET.NE.0) GOTO 53
      ISSS=ISS  /*NEW: STARTING SENSOR - SAVE FOR 2ND PASS
      ISENS=ISS
   53 IF (ISS.NE.ISENS) GOTO 900
      IF (LINE.NE.LSTART .AND. LINE.NE.LPI) GOTO 60
      ISP=256*RS(IBUFF(IS),8)+RT(IBUFF(IS),8)
      IFP=256*RS(IBUFF(IF),8)+RT(IBUFF(IF),8)
      IF (LINE.NE.LSTART) GOTO 54
      IST=ISP
      IFIN=IFP
   54 IF (LINE.NE.LPI) GOTO 60
      IST2=ISP
      IFIN2=IFP
   60 IF (NEW) GOTO 80
      IF (LINE.LT.LSTART) GOTO 80
      IF (NBN(IBAND).NE.1) GOTO 80  /*USE ONLY SPECIFIED BANDS
      IF (LINET.GE.NLINES) GOTO 80  /*CHECK NLINES ONLY
      LINET=LINET+1
      DO 64 K=1,NWB  /*UNPACK BYTES TO WORDS...
      KA=NWB1-K     /*...BACKWARDS,...
      KB=KA+KA       /*...USING THE SAME BUFFER
      IW=IBUFF(KA)
      IBUFF(KB-1)=RS(IW,8)
   64 IBUFF(KB)=RT(IW,8)
      J1=ISP
      IF (IBAND.NE.1) J1=J1-170
C "OLD" TAPES: ISTORE=1 FOR PIX VALUES NOT IN LOOK-UP TABLE
      ITEST=IFP-ISP+101  /*NO OF PIX IN BAND ? LINE
      KA=J1+5-2*(IBAND-1)
      DO 70 K=1,ITEST
      IPIX=IBUFF(K+KA)  /*START AT 7TH 5TH 3RD OR 1ST PIX
      DO 68 L=1,64
      IF (NOS(IBAND,ISENS,L).EQ.IPIX) GOTO 70
   68 CONTINUE
      ISTORE(K)=1
   70 CONTINUE
   80 CONTINUE
  100 CONTINUE
C
      CALL REW(IMUNIT)
      IF (NEW) GOTO 120
      WRITE(1,1104)
 1104 FORMAT('ADDED PIXELS AT POSITIONS:-')
      IOLD=0
      ITESTL=ITEST+6  /*POSSIBLE ISTORE VALUES
      IF (NEW) ITESTL=ITEST
      DO 110 I=1,ITESTL
      IF (ISTORE(I).EQ.0) GOTO 110
      IDIFF=I-IOLD
      WRITE(1,1105)I,IDIFF
 1105 FORMAT(2I6)
      IOLD=I
  110 CONTINUE
      I=ITESTL-IOLD
```

```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


      WRITE(1,1107)I
 1107 FORMAT(6X,I6)
C
  120 IF (IST2.GT.IST) IST=IST2
      IF (IFIN2.LT.IFIN) IFIN=IFIN2
      IF (.NOT.NEW) IEPL=IFIN-IST+101
      IF (NEW) IEPL=IFIN-IST-5
      WRITE(1,1106)NLINE,IEPL
 1106 FORMAT('IMAGE SIZE:',I5,' ROWS BY',I5,' COLS')
C
C OPEN OUTPUT IMAGE FILES USING TEXT AND NAME OBTAINED EARLIER AND IEPL
      ICHTEX=NLENSA(ITEXT,80)+1  /*PDR 3106 P23-6
      ICHNAM=NLENSA(NAME,32)+1  /*FIRST FREE LOCATION
      CALL INITI
      DO 130 I=1,4
      IF (NBN(I).NE.1) GOTO 130
      CALL NCHRSA(ITEXT,ICHTEX,SUFFIX,I) /*ADD SUFFIX TO MESSAGE
      CALL NCHRSA(NAME,ICHNAM,SUFFIX,I)  /*ADD SUFFIX TO FILENAME
      CALL OIWX(I,'I1',NAME,IEPL,NLINE,ITEXT)
  130 CONTINUE
C
C ****** NOW FOR SECOND PASS ******
      PRINTH=.FALSE.
      CALL LHEADR(IMUNIT,PRINTH)  /*NO OUTPUT TO VDU ON SECOND PASS
C
      ISENS=0
      IF (NEW) ISENS=ISSS-1
      DO 200 LINE=1,LPI
      ISENS=ISENS+1
      IF (ISENS.GT.6) ISENS=1
      DO 190 IBAND=1,4
      CALL MTRE(IMUNIT,IBUFF,NWB,EOT,EOF,ERR,NN)
      IF (EOF) GOTO 990
      IF (LINE.LT.LSTART) GOTO 190
      IF (IBAND.EQ.1) ISP=256*RS(IBUFF(IS),8)+RT(IBUFF(IS),8)
      IF (NBN(IBAND).NE.1) GOTO 190
      DO 164 K=1,NWB  /*UNPACK BYTES TO WORDS...
      KA=NWB1-K     /*...BACKWARDS,...
      KB=KA+KA       /*...USING THE SAME BUFFER
      IW=IBUFF(KA)
      IBUFF(KB-1)=RS(IW,8)
  164 IBUFF(KB)=RT(IW,8)
C TRANSFER THE LINE, ALTERING ADDED PIX AND UNCORRECTING
      KSHIFT=IST-1  /*FOR IBUFF POINTER
      IF (IBAND.NE.1) KSHIFT=KSHIFT-170
      KIS=IST-ISP-(6-2*(IBAND-1))
      ITEMP=-1
      FLAG=.FALSE.
      DO 170 K=1,IEPL
      IF (NEW) GOTO 166
      IF (ISTORE(K+KIS).NE.1) GOTO 166
      IF (ITEMP.NE.-1) GOTO 169
      FLAG=.TRUE.
      GOTO 170
  166 ITEMP=IBUFF(K+KSHIFT)+1
      ITEMP=NBAK(IBAND,ISENS,ITEMP)  /*UN-CORRECT THE PIX
      IF (ITEMP.EQ.-1) GOTO 970  /*VALUE NOT IN LOOK-UP TABLE
  168 IF (.NOT.FLAG) GOTO 169
      IBUFF(K-1)=ITEMP
      FLAG=.FALSE.
```

018

```
 C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


   169 IBUFF(K)=ITEMP
   170 CONTINUE
       CALL WLI2(IBAND,IBUFF)  /*WRITE OUTPUT LINE TO FILE
   190 CONTINUE
   200 CONTINUE
 C
       CALL REW(IMUNIT)
       DO 250 I=1,4
       IF (NBN(I).NE.1) GOTO 250
       CALL CLOSEI(I)
   250 CONTINUE
       STOP :123456
 C
   900 WRITE(1,1049)ISS,ISENS
  1049 FORMAT('FAILS SENSOR CHECK. ISS, ISENS:',2I6)
       GOTO 995
 C
   970 IB=IBAND+3
       IF (.NOT.NEW) WRITE(1,1070)LINE,IB,K,IPIX
  1070 FORMAT('LINE',I5,', BAND',I2,', PIXEL NO',I5,', VALUE',I6,
      * ' NOT IN TABLE')
 C      GOTO 995
       GOTO 168  /*TEMPORARY
 C
   980 WRITE(1,1080)
  1080 FORMAT('ERROR 980')
       GOTO 995
 C
   990 WRITE(1,1090)
  1090 FORMAT('DOUBLE FILE MARK REACHED')
   995 CALL REW(IMUNIT)
       CALL CALLI
       STOP :7777
       END
C*****************************************************************************
       SUBROUTINE OIWX(FUNIT,ET,NAME,EPL,LPI,TEXT)
       INTEGER FUNIT,ET,EPL,LPI,TEXT(40)
       DIMENSION NAME(16)
       INTEGER EXTRA(44)
       INTEGER STATE(16),START(16),CET(16),CEPL(16)
       INTEGER CLPI(16),LNUM(16)
       COMMON/IMCOM/STATE,START,CET,CEPL,CLPI,LNUM
C
C
C CHECK UNIT IS CLOSED
       IF(STATE(FUNIT).EQ.0) GOTO 4
       WRITE(1,400) FUNIT
400    FORMAT('*** ERROR ***'/
      * 'ATTEMPT TO OPEN IMAGE FOR WRITING ON UNIT ',I2/
      * 'WHEN UNIT IN USE')
       CALL CALLI
       CALL EXIT
4      CONTINUE
C CHECK EPL .GE. 0 AND THAT LPI .GE. 0
       IF(EPL.GE.1 .AND. LPI .GE. 1) GOTO 20
       WRITE(1,100)FUNIT
100    FORMAT('*** ERROR ***'/
      * 'ATTEMPT TO OPEN IMAGE FOR WRITING ON UNIT ',I2/
      * 'WITH EPL OR LPI LESS THAN ONE')
       CALL CALLI
```

018

```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


      CALL EXIT
20    CONTINUE
C OPEN FILE AND SET EXTRA INFO
3     CALL ODAMUX(FUNIT,NAME)
      START(FUNIT) = 44
      CET(FUNIT) = ET
      CEPL(FUNIT) = EPL
      CLPI(FUNIT) = LPI
      EXTRA(1) = START(FUNIT)
      EXTRA(2) = CET(FUNIT)
      EXTRA(3) = CEPL(FUNIT)
      EXTRA(4) = CLPI(FUNIT)
C
      DO 1 I=1,40
      EXTRA(I + 4) = TEXT(I)
1     CONTINUE
      CALL POSN(FUNIT,0000000)
      CALL WTDAM(FUNIT,EXTRA,44)
      LNUM(FUNIT) = 1
      STATE(FUNIT) = 2    /*  OPEN FOR WRITING
      RETURN
      END
C**************************************************************************
      SUBROUTINE ODAMUX(FUNIT,NAME)
C OPENS A DAM FILE FOR WRITING
$INSERT SYSCOM>A$KEYS
      INTEGER FUNIT
      LOGICAL OK
      INTEGER NAME(16)
C
C
1     OK = OPEN$A(A$WRIT + A$DAMF,NAME,32,FUNIT)
      IF(.NOT.OK) CALL EXIT
      CALL POSN(FUNIT, 0000000)
      RETURN
      END
C*********************************************************************
      SUBROUTINE LHEADR(IUNIT,PRINTH)
C READS HEADER BLOCKS.  IF PRINTH, OUTPUTS DETAILS AND ASKS QUESTIONS
      INTEGER BUFF(1530)  /*LARGE ENOUGH FOR JSC HEADER
      LOGICAL EOT,EOF,ERR,ASCII,NEW,PRINTH
C
C
      CALL MTRE(IUNIT,BUFF,1530,EOT,EOF,ERR,NW)  /*READ BLOCK 1
      IF (PRINTH) CALL JSCPT(BUFF)  /*WRITE JSC HDR TO VDU
      CALL MTRE(IUNIT,BUFF,1530,EOT,EOF,ERR,NW) /*READ EOF
      IF (.NOT.EOF) WRITE(1,1002)
 1002 FORMAT('NOT EOF FOLLOWING JSC HEADER')
      CALL MTRE(IUNIT,BUFF,720,EOT,EOF,ERR,NW) /*LANDSAT HDR
      CALL LANPT(BUFF,ASCII,NEW,PRINTH)  /*LANDSAT HEADER
      CALL MTRE(IUNIT,BUFF,360,EOT,EOF,ERR,NW) /*TRANSFORMATION RECORD
      IF (PRINTH) CALL TRANPT(BUFF,ASCII)  /*WRITE TRANSFORMATION RECORD
      DO 30 J=4,8  /*DATA RECORDS 4 TO 8
      CALL MTRE(IUNIT,BUFF,810,EOT,EOF,ERR,NW)  /*LOOK-UP RECORDS
      IF (PRINTH.AND.J.NE.8) CALL LOOKPT(BUFF,J,ASCII)  /*WRITE A TABLE
   30 CONTINUE
      CALL MTRE(IUNIT,BUFF,1530,EOT,EOF,ERR,NW) /*READ EOF
      IF (.NOT.EOF) WRITE(1,1003)
 1003 FORMAT('NOT EOF AFTER LOOK-UP TABLES')
      RETURN
```

018

PAGE 0007

```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


      END
C**********************************************************************
C SUBROUTINES TO PRINT HEADER RECORDS FROM FUCINO LANDSAT TAPES
C
      SUBROUTINE JSCPT(IPT)
C SUBROUTINE TO PRINT THE J.S.C. HEADER (3060 BYTES). ALWAYS EBCDIC CHARS.
      DIMENSION IPT(1),IB(17)
C
C
      IB(1)=LS(RT(IPT(1369),8),8)  /*1ST BYTE
      DO 10 I=1,8   /*NEXT 16 BYTES
      IC1=2*I
      IC2=IC1+1
      IB(IC1)=LT(IPT(I+1369),8)
   10 IB(IC2)=LS(RT(IPT(I+1369),8),8)
C
      CALL EBCASC(IB,16,1)  /*CONVERT TO ASCII, UPPER CASE
      CALL EBCASC(IPT(1444),8,1)
      WRITE(1,1000) (IB(I),I=1,16),IPT(1443),(IPT(I),I=1444,1447,
 1000 FORMAT(///'*********  JSC HEADER  **********'//
     1 'SUN ELEVATION  :',8A1,' MILLIRADS'//
     2 'SUN AZIMUTH    :',8A1,' MILLIRADS'//
     3 'EARTH ROTATION :',I8,' MILLIRADS'//
     4 'SATELLITE ALT  :',4A2,' METRES'//)
      RETURN
      END
C
C**********************************************************************
      SUBROUTINE LANPT(IPT,ASCII,NEW,PRINTH)
C DETERMINE ASCII, NEW.  IF REQD, PRINT LANDSAT HEADER (1440 BYTES)
      LOGICAL ASCII,NEW,PRINTH
      DIMENSION IPT(1)
C
C
      ASCII=.TRUE.
      I=RT(IPT(605),8)  /*FLAG FOR ASCII/EBCDIC
      IF (I.EQ.240) ASCII=.FALSE.  /*240=360 OCTAL =EBCDIC ZERO
      IF (.NOT.ASCII) CALL EBCASC(IPT(1),720,8)  /*CONVERT TO ASCII
      M=IPT(484)
      M1=RS(M,8)  /*FIRST BYTE
      M1=RT(M1,4)  /*KEEPS 4 BITS ONLY
      M2=RT(M,4)  /*SECOND BYTE, 4 CHARS ONLY
      M=10*M1+M2  /*ASSEMBLE AS A NUMBER
      IYR=IPT(485)
      I1=RS(IYR,8)
      I1=RT(I1,4)
      I2=RT(IYR,4)
      IYR=10*I1+I2
      NEW=.FALSE.
      IF (IYR.GT.79) NEW=.TRUE.
      IF (IYR.EQ.79 .AND. M.GE.8) NEW=.TRUE.
      IF (PRINTH) WRITE(1,1000) (IPT(I),I=1,720)
 1000 FORMAT(/'********  LANDSAT HEADER  ********',/18(40A2/)/)
      RETURN
      END
C**********************************************************************
      SUBROUTINE TRANPT(IPT,ASCII)  /*PRINT TRANSFORMATION RECORD
      LOGICAL ASCII
      DIMENSION IPT(1)
C
```

```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


C
      IF (.NOT.ASCII) CALL EBCASC(IPT(1),360,0)
      WRITE(1,1000)(IPT(I),I=1,360)
 1000 FORMAT(/'******** TRANSFORMATION RECORD *********'/9(/40A2))
      RETURN
      END
C*********************************************************************
      SUBROUTINE LOOKPT(IPT,NT,ASCII)  /*PRINTS RADIOMETRIC LOOK-UP TABLE NT
      LOGICAL ASCII
      DIMENSION IPT(1)
C
C
      IF (.NOT.ASCII) CALL EBCASC(IPT(1),768,0)
      M=768
      IF (NT.EQ.8) M=256  /*THERMAL BAND 8 HAS 2 SENSORS
      WRITE(1,1000)NT,(IPT(J),J=1,M)
 1000 FORMAT(//' RADIOMETRIC LOOK-UP TABLE ****** BAND',I2,
     * 6(/4(/32A2)))
      RETURN
      END
C*********************************************************************
      SUBROUTINE EBCASC(IWORD,N,IUP)  /*CONVERTS N CHARS, EBCDIC TO ASCII
C STARTING FROM IWORD
C                    )
      INTEGER EBTOAS,EBCHAR,ASCHAR
      INTEGER IWORD(1),LCHAR(2),J,M,ICHAR,L
      INTEGER EBTOAS(27)
      DATA EBTOAS/:006615,:040240,:055241,:077642,:075643,:045244,
     : :066245,:050246,:076647,:046650,:056651,:056252,:047253,:065654,
     : :060255,:045656,:060657,:075272,:056673,:046274,:077275,:067276,
     : :067677,:076300,:066737,:022612,:055644/
C
C
      IF (N .LE. 0)GOTO 150
      DO 10 J=1,N
C
C     SPLIT IWORD(J) INTO TWO SINGLE CHARACTERS LCHAR(1) & LCHAR(2)
C
      LCHAR(1)= RS((AND(IWORD(J),:177400)),8)
      LCHAR(2)= AND(IWORD(J),:377)
      DO 20 M=1,2
      ICHAR=LCHAR(M)
      L=1
C
C     SEE WHETHER ICHAR IS ALPHANUMERIC
C
      IF (ICHAR .LT. :360 .OR. ICHAR .GT. :371)GOTO 200
      ICHAR= ICHAR-:100
      GOTO 120
  200 IF (ICHAR .LT. :301 .OR. ICHAR .GT. :311)GOTO 210
      GOTO 120
  210 IF (ICHAR .LT. :321 .OR. ICHAR .GT. :331)GOTO 220
      ICHAR= ICHAR-:007
      GOTO 120
  220 IF (ICHAR .LT. :342 .OR. ICHAR .GT. :351)GOTO 230
      ICHAR= ICHAR-:017
      GOTO 120
  230 IF (ICHAR .LT. :201 .OR. ICHAR .GT. :211)GOTO 240
      ICHAR= ICHAR+:140
      GOTO 130
```

018

```
C PROGRAM IM.UNCOR: TO UNCORRECT AN OLD OR NEW CORRECTED FUCINO TAPE


  240 IF (ICHAR .LT. :221 .OR. ICHAR .GT. :231)GOTO 250
      ICHAR= ICHAR+:131
      GOTO 130
  250 IF (ICHAR .LT. :242 .OR. ICHAR .GT. :251)GOTO 260
      ICHAR= ICHAR+:121
      GOTO 130
C
C     SEE WHETHER ICHAR IS IN THE ARRAY EBTOAS
C
  260 EBCHAR= RS((AND(EBTOAS(L),:177400)),8)
      IF (ICHAR .EQ. EBCHAR) GOTO 40
      L=L+1
      IF (L .LE. 27)GOTO 260
      GOTO 50
   40 ASCHAR= AND(EBTOAS(L),:377)
      ICHAR = ASCHAR
      GOTO 120
C
C     SET ICHAR TO NUL CHARACTER
C
   50 ICHAR = :200
      GOTO 120
C
C     IF IUP = 1 UPCASE ICHAR
C
  130 IF (IUP .NE. 1)GOTO 120
      ICHAR= ICHAR-:040
  120 LCHAR(M)= ICHAR
   20 CONTINUE
C
C     RECOMBINE LCHAR(1) AND LCHAR(2) INTO A SINGLE WORD (IWORD(J))
C
      IWORD(J)= OR(LS(LCHAR(1),8),LCHAR(2))
   10 CONTINUE
  150 RETURN
      END
```

Appendix B

LISTING OF PROGRAM IM.FUC.MSS.2


```
C PROGRAM IM.FUC.MSS.2


C PROGRAM IM.FUC.MSS.2
C TO READ A FUCINO MAGTAPE, IN OLD OR NEW FORMAT, TO IMAGE-FILES
      PARAMETER M=3780  /*NO. OF BYTES PER IMAGE BLOCK
      PARAMETER NWB=1890  /*NO. OF WORDS PER IMAGE BLOCK
      LOGICAL EOT,EOF,ERR,Q,NEW,PRINTH
      INTEGER*4 DNUM
      DIMENSION IBUFW(NWB),IBUFB(M),ISTORE(M),ITEXT(40),NBN(4)
      DIMENSION NAME(16)
$INSERT SYSCOM>A$KEYS
      DATA SUFFIX /4H4567/
C
C
C
      WRITE(1,1000)
 1000 FORMAT('IM.FUC.MSS.2   11-JAN-80')
C
   20 Q=RNUM$A('INPUT TAPE UNIT NUMBER',22,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 20
      IMUNIT=INTS(DNUM)
      IF (IMUNIT.LT.0 .OR. IMUNIT.GT.3) GOTO 20
      CALL REW(IMUNIT)
C
      DO 30 J=1,4
   30 NBN(J)=0  /*NO BAND IN USE..
      Q=YSNO$A('ALL BANDS',9,A$NDEF)
      IF (.NOT.Q) GOTO 33
      DO 32 J=1,4
   32 NBN(J)=1
      NBANDS=4
      GOTO 40
   33 Q=RNUM$A('HOW MANY BANDS',14,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 33
      NBANDS=INTS(DNUM)
      IF (NBANDS.LT.1 .OR.NBANDS.GT.3) GOTO 33
      DO 39 J=1,NBANDS
   35 Q=RNUM$A('BAND NO',7,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 35
      JB=INTS(DNUM)
      IF (JB.GE.4 .AND. JB.LE.7) GOTO 38
   37 WRITE(1,1002)
 1002 FORMAT('REPEAT THAT NUMBER')
      GOTO 35
   38 JC=JB-3
      IF (NBN(JC).EQ.1) GOTO 37
      NBN(JC)=1
   39 CONTINUE
C
   40 LSTART=1  /*START FOR ENTIRE IMAGE
      NLINE=2286  /*NO. OF LINES IN ENTIRE IMAGE
      LPI=NLINE  /*END LINE FOR ENTIRE IMAGE
      Q=YSNO$A('ENTIRE IMAGE',12,A$NDEF)
      IF (Q) GOTO 50
   42 Q=RNUM$A('STARTING LINE',13,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 42
      LSTART=INTS(DNUM)
      IF (LSTART.LT.1 .OR.LSTART.GT.2286) GOTO 42
   44 Q=RNUM$A('HOW MANY LINES',14,A$DEC,DNUM)
      IF (.NOT.Q) GOTO 44
      NLINE=INTS(DNUM)
      IF (NLINE.LT.0 .OR. NLINE.GT.2286) GOTO 44
```

```
C PROGRAM IM.FUC.MSS.2


      IF (NLINE.EQ.0) GOTO 60  /*FOR HEADER PRINTOUT ONLY
      LPI=LSTART+NLINE-1  /*LAST LINE OF OUTPUT IMAGE
      IF (LPI.GT.2286) GOTO 42
C
C READ TEXT AND FILENAME FROM OPERATOR
   50 WRITE(1,1004)
 1004 FORMAT('ENTER TEXT MESSAGE FOR IMAGE')
      READ(1,1005)ITEXT
 1005 FORMAT(40A2)
   52 Q=RNAM$A('OUTPUT FILE NAME',16,A$FUPP,NAME,32)
      IF (.NOT.Q) GOTO 52
C
C READ THROUGH HEADERS, WITH PRINTOUT
   60 PRINTH=.TRUE.  /*PRINT THE HEADER
      CALL LHEADR(INUNIT,PRINTH,NEW)  /*-1 FOR PRINTOUT
      IF (NLINE.EQ.0) STOP :12345  /*NLINE=0 FOR HEADER PRINT ONLY
C
C READ TAPE ONCE TO DETERMINE MAXIMUM START AND FINISH COLUMNS
      DO 100 LINE=1,LPI
      DO 90 IBAND=1,4
      CALL MTRE(INUNIT,IBUFW,NWB,EOT,EOF,ERR,NW)
      IF (EOF) GOTO 990
      IF (IBAND.NE.1) GOTO 90
      IF (LINE.NE.LSTART) GOTO 70
      IST=256*RS(IBUFW(54),8)+RT(IBUFW(54),8)
      IFIN=256*RS(IBUFW(55),8)+RT(IBUFW(55),8)
   70 IF (LINE.NE.LPI) GOTO 90
      IST2=256*RS(IBUFW(54),8)+RT(IBUFW(54),8)
      IFIN2=256*RS(IBUFW(55),8)+RT(IBUFW(55),8)
   90 CONTINUE
  100 CONTINUE
      CALL REW(INUNIT)
      IF (IST2.GT.IST) IST=IST2
      IF (IFIN2.LT.IFIN) IFIN=IFIN2
      IF (NEW) IEPL=IFIN-IST-5  /*NO. OF PIX IN LINE FOR OUTPUT IMAGE
      IF (.NOT.NEW) IEPL=IFIN-IST+181
C
C OPEN IMAGE FILES FOR WRITING USING TEXT AND NAME OBTAINED EARLIER
      CALL INITI
      ICHTEX=NLEN$A(ITEXT,80)+1  /*PDR 3106 P23-6
      ICHNAM=NLEN$A(NAME,32)+1  /*POSITION FOR SUFFIX
      DO 105 IBAND=1,4
      IF (NBN(IBAND).NE.1) GOTO 105
      CALL MCHR$A(ITEXT,ICHTEX,SUFFIX,IBAND)  /*PDR 3106 P23-7
      CALL MCHR$A(NAME,ICHNAM,SUFFIX,IBAND)  /*ADDS SUFFIX
      CALL OIUX(IBAND,'I1',NAME,IEPL,NLINE,ITEXT)
  105 CONTINUE
      WRITE(1,1006)NLINE,IEPL
 1006 FORMAT('IMAGE SIZE:',I5,' ROWS',I5,' COLUMNS')
C
      PRINTH=.FALSE.
      CALL LHEADR(INUNIT,PRINTH,NEW)  /*NO OUTPUT TO VDU ON SECOND PASS
C
      DO 200 LINE=1,LPI
      DO 190 IBAND=1,4
      CALL MTRE(INUNIT,IBUFW,NWB,EOT,EOF,ERR,NN)
      IF (EOF) GOTO 990
      IF (LINE.LT.LSTART) GOTO 190
      IF (NBN(IBAND).NE.1) GOTO 190  /*IGNORE THIS BAND
      DO 110 K=1,NWB
```

```
C PROGRAM IM.FUC.MSS.2


      KK=K+K-1
      IBUFB(KK)=RS(IBUFW(K),8)
  110 IBUFB(KK+1)=RT(IBUFW(K),8)
      KSHIFT=IST-1
      IF (IBAND.NE.1) KSHIFT=KSHIFT-178
      DO 130 K=1,IEPL
      ISTORE(K)=IBUFB(K+KSHIFT)
  130 CONTINUE
      CALL WLI2(IBAND,ISTORE)
  190 CONTINUE
  200 CONTINUE
C
      CALL REW(IMUNIT)
      DO 250 I=1,4
      IF (NBN(I).NE.1) GOTO 250
      CALL CLOSEI(I)
  250 CONTINUE
      STOP :123456
C
  990 WRITE(1,1090)
 1090 FORMAT('DOUBLE FILE MARK REACHED')
      CALL REW(IMUNIT)
      STOP :7777
      END
C**************************************************************************************
      SUBROUTINE OIWX(FUNIT,ET,NAME,EPL,LPI,TEXT)
      INTEGER FUNIT,ET,EPL,LPI,TEXT(40)
      DIMENSION NAME(16)
      INTEGER EXTRA(44)
      INTEGER STATE(16),START(16),CET(16),CEPL(16)
      INTEGER CLPI(16),LNUM(16)
      COMMON/IMCOM/STATE,START,CET,CEPL,CLPI,LNUM
C
C
C CHECK UNIT IS CLOSED
      IF(STATE(FUNIT).EQ.0) GOTO 4
      WRITE(1,400) FUNIT
400   FORMAT('*** ERROR ***'/
     * 'ATTEMPT TO OPEN IMAGE FOR WRITING ON UNIT ',I2/
     * 'WHEN UNIT IN USE')
      CALL CALLI
      CALL EXIT
4     CONTINUE
C CHECK EPL .GE. 0 AND THAT LPI .GE. 0
      IF(EPL.GE.1 .AND. LPI .GE. 1) GOTO 20
      WRITE(1,100)FUNIT
100   FORMAT('*** ERROR ***'/
     * 'ATTEMPT TO OPEN IMAGE FOR WRITING ON UNIT ',I2/
     * 'WITH EPL OR LPI LESS THAN ONE')
      CALL CALLI
      CALL EXIT
20    CONTINUE
C OPEN FILE AND SET EXTRA INFO
3     CALL ODAMWX(FUNIT,NAME)
      START(FUNIT) = 44
      CET(FUNIT) = ET
      CEPL(FUNIT) = EPL
      CLPI(FUNIT) = LPI
      EXTRA(1) = START(FUNIT)
      EXTRA(2) = CET(FUNIT)
```

PAGE 0004

```
C PROGRAM IM.FUC.MSS.2


      EXTRA(3) = CEPL(FUNIT)
      EXTRA(4) = CLPI(FUNIT)
C
      DO 1 I=1,40
      EXTRA(I + 4) = TEXT(I)
1     CONTINUE
      CALL POSN(FUNIT,0000000)
      CALL WTDAM(FUNIT,EXTRA,44)
      LNUM(FUNIT) = 1
      STATE(FUNIT) = 2    /*  OPEN FOR WRITING
      RETURN
      END
C*****************************************************************************
      SUBROUTINE ODAMWX(FUNIT,NAME)
C OPENS A DAM FILE FOR WRITING
$INSERT SYSCOM>A$KEYS
      INTEGER FUNIT
      LOGICAL OK
      INTEGER NAME(16)
C
C
1     OK = OPEN$A(A$WRIT + A$DAMF,NAME,32,FUNIT)
      IF(.NOT.OK) CALL EXIT
      CALL POSN(FUNIT, 0000000)
      RETURN
      END
C**********************************************************************
      SUBROUTINE LHEADR(IUNIT,PRINTH,NEW)
C READS HEADER BLOCKS.  IF PRINTH IS TRUE, PRINTS DETAILS
      INTEGER BUFF(1530)  /*LARGE ENOUGH FOR JSC HEADER
      LOGICAL EOT,EOF,ERR,ASCII,NEW,PRINTH
C
C
      CALL MTRE(IUNIT,BUFF,1530,EOT,EOF,ERR,NW)  /*READ BLOCK 1
      IF (PRINTH) CALL JSCPT(BUFF)  /*WRITE JSC HDR TO VDU
      CALL MTRE(IUNIT,BUFF,1530,EOT,EOF,ERR,NW) /*READ EOF
      IF (.NOT.EOF) WRITE(1,1002)
 1002 FORMAT('NOT EOF FOLLOWING JSC HEADER')
      CALL MTRE(IUNIT,BUFF,720,EOT,EOF,ERR,NW) /*LANDSAT HDR
      CALL LANPT(BUFF,ASCII,NEW,PRINTH)  /*DETERMINE ASCII, NEW
      CALL MTRE(IUNIT,BUFF,360,EOT,EOF,ERR,NW) /*TRANSFORMATION RECORD
      IF (PRINTH) CALL TRANPT(BUFF,ASCII)  /*WRITE TRANSFORMATION RECORD
      DO 30 J=4,8  /*DATA RECORDS 4 TO 8
      CALL MTRE(IUNIT,BUFF,810,EOT,EOF,ERR,NW)  /*LOOK-UP RECORDS
      IF (PRINTH.AND.J.NE.8) CALL LOOKPT(BUFF,J,ASCII)  /*WRITE A TABLE
   30 CONTINUE
      CALL MTRE(IUNIT,BUFF,1530,EOT,EOF,ERR,NW) /*READ EOF
      IF (.NOT.EOF) WRITE(1,1003)
 1003 FORMAT('NOT EOF AFTER LOOK-UP TABLES')
      RETURN
      END
C**********************************************************************
      SUBROUTINE JSCPT(IPT)
C SUBROUTINE TO PRINT THE J.S.C. HEADER (3060 BYTES). ALWAYS EBCDIC CHARS.
      DIMENSION IPT(1),IB(17)
C
C
      IB(1)=LS(RT(IPT(1369),8),8)  /*1ST BYTE
      DO 10 I=1,8   /*NEXT 16 BYTES
      IC1=2*I
```

```
C PROGRAM IM.FUC.MSS.2

      IC2=IC1+1
      IB(IC1)=LT(IPT(I+1369),8)
   10 IB(IC2)=LS(RT(IPT(I+1369),8),8)
C
      CALL EBCASC(IB,16,1)  /*CONVERT TO ASCII, UPPER CASE
      CALL EBCASC(IPT(1444),8,1)
      WRITE(1,1000) (IB(I),I=1,16),IPT(1443),(IPT(I),I=1444,1447)
 1000 FORMAT(///'*********  JSC HEADER  **********'//
     1 'SUN ELEVATION   :',8A1,' MILLIRADS'//
     2 'SUN AZIMUTH     :',8A1,' MILLIRADS'//
     3 'EARTH ROTATION  :',I8,' MILLIRADS'//
     4 'SATELLITE ALT   :',4A2,' METRES'//)
      RETURN
      END
C
C*************************************************************************
      SUBROUTINE LANPT(IPT,ASCII,NEW,PRINTH)
C DETERMINE ASCII, NEW.  IF REQD, PRINT LANDSAT HEADER (1440 BYTES)
      LOGICAL ASCII,NEW,PRINTH
      DIMENSION IPT(1)
      DOUBLE PRECISION CHASC,CHEBC
      DATA CHASC,CHEBC/8HASCII   ,8HEBCDIC
C
C
      I=RT(IPT(685),8)  /*FLAG FOR ASCII/EBCDIC
      IF (I.NE.240) GOTO 20  /*IT'S NOT EBCDIC (240=360 OCTAL=EBCDIC 0
      ASCII=.FALSE.
      CALL EBCASC(IPT(1),720,0)  /*CONVERT TO ASCII
      IF (PRINTH) WRITE(1,1001)CHEBC
 1001 FORMAT('MAGTAPE CHARACTERS ARE ',A8)
      GOTO 30
   20 ASCII=.TRUE.
      IF (PRINTH) WRITE(1,1001)CHASC
   30 M=IPT(484)
      M1=RS(M,8)  /*FIRST BYTE
      M1=RT(M1,4)  /*KEEPS 4 BITS ONLY
      M2=RT(M,4)  /*SECOND BYTE, 4 CHARS ONLY
      M=10*M1+M2  /*ASSEMBLE AS A NUMBER
      IYR=IPT(485)
      I1=RS(IYR,8)
      I1=RT(I1,4)
      I2=RT(IYR,4)
      IYR=10*I1+I2
      NEW=.FALSE.
      IF (IYR.GT.79) NEW=.TRUE.
      IF (IYR.EQ.79 .AND. M.GE.8) NEW=.TRUE.
      IF (IACT.LT.0) WRITE(1,1000) (IPT(I),I=1,720)
 1000 FORMAT(/'********  LANDSAT HEADER  ********',/18(40A2/)/)
      RETURN
      END
C*************************************************************************
      SUBROUTINE TRANPT(IPT,ASCII)  /*PRINT TRANSFORMATION RECORD
      LOGICAL ASCII
      DIMENSION IPT(1)
C
C
      IF (.NOT.ASCII) CALL EBCASC(IPT(1),360,0)
      WRITE(1,1000)(IPT(I),I=1,360)
 1000 FORMAT(/'******** TRANSFORMATION RECORD *********'/9(/40A2))
      RETURN
```

```
C PROGRAM IM.FUC.MSS.2


      END
C**********************************************************************
      SUBROUTINE LOOKPT(IPT,NT,ASCII)  /*PRINTS RADIOMETRIC LOOK-UP TABLE NT
      LOGICAL ASCII
      DIMENSION IPT(1)
C
C
      IF (.NOT.ASCII) CALL EBCASC(IPT(1),810,0)
      M=768
      IF (NT.EQ.8) M=256  /*THERMAL BAND 8 HAS 2 SENSORS
      WRITE(1,1000)NT,(IPT(J),J=1,M)
 1000 FORMAT(//' RADIOMETRIC LOOK-UP TABLE ****** BAND',I2,
     * 6(/4(/32A2)))
      RETURN
      END
C**********************************************************************
      SUBROUTINE EBCASC(IWORD,N,IUP)  /*CONVERTS N CHARS, EBCDIC TO ASCII
C STARTING FROM IWORD
C
      INTEGER EBTOAS,EBCHAR,ASCHAR
      INTEGER IWORD(1),LCHAR(2),J,M,ICHAR,L
      INTEGER EBTOAS(27)
      DATA EBTOAS/:006615,:040240,:055241,:077642,:075643,:045244,
     ! :066245,:050246,:076647,:046650,:056651,:056252,:047253,:065654,
     ! :060255,:045656,:060657,:075272,:056673,:046274,:077275,:067276,
     ! :067677,:076300,:066737,:022612,:055644/
C
C
      IF (N .LE. 0)GOTO 150
      DO 10 J=1,N
C
C     SPLIT IWORD(J) INTO TWO SINGLE CHARACTERS LCHAR(1) & LCHAR(2)
C
      LCHAR(1)= RS((AND(IWORD(J),:177400)),8)
      LCHAR(2)= AND(IWORD(J),:377)
      DO 20 M=1,2
      ICHAR=LCHAR(M)
      L=1
C
C     SEE WHETHER ICHAR IS ALPHANUMERIC
C
      IF (ICHAR .LT. :360 .OR. ICHAR .GT. :371)GOTO 200
      ICHAR= ICHAR-:100
      GOTO 120
  200 IF (ICHAR .LT. :301 .OR. ICHAR .GT. :311)GOTO 210
      GOTO 120
  210 IF (ICHAR .LT. :321 .OR. ICHAR .GT. :331)GOTO 220
      ICHAR= ICHAR-:007
      GOTO 120
  220 IF (ICHAR .LT. :342 .OR. ICHAR .GT. :351)GOTO 230
      ICHAR= ICHAR-:017
      GOTO 120
  230 IF (ICHAR .LT. :201 .OR. ICHAR .GT. :211)GOTO 240
      ICHAR= ICHAR+:140
      GOTO 130
  240 IF (ICHAR .LT. :221 .OR. ICHAR .GT. :231)GOTO 250
      ICHAR= ICHAR+:131
      GOTO 130
  250 IF (ICHAR .LT. :242 .OR. ICHAR .GT. :251)GOTO 260
      ICHAR= ICHAR+:121
```

PAGE 0007

```
C PROGRAM IN.FUC.NSS.2


      GOTO 130
C
C     SEE WHETHER ICHAR IS IN THE ARRAY EBTOAS
C
  260 EBCHAR= RS((AND(EBTOAS(L),:177400)),8)
      IF (ICHAR .EQ. EBCHAR) GOTO 40
      L=L+1
      IF (L  .LE. 27)GOTO 260
      GOTO 50
   40 ASCHAR= AND(EBTOAS(L),:377)
      ICHAR = ASCHAR
      GOTO 120
C
C     SET ICHAR TO NUL CHARACTER
C
   50 ICHAR = :200
      GOTO 120
C
C     IF IUP = 1 UPCASE ICHAR
C
  130 IF (IUP .NE. 1)GOTO 120
      ICHAR= ICHAR-:040
  120 LCHAR(N)= ICHAR
   20 CONTINUE
C
C     RECOMBINE LCHAR(1) AND LCHAR(2) INTO A SINGLE WORD (IWORD(J))
C
      IWORD(J)= OR(LS(LCHAR(1),8),LCHAR(2))
   10 CONTINUE
  150 RETURN
      END
```

Table 1

RECORD STRUCTURE OF LANDSAT MSS CCT

Data record	Number of bytes
JSC header	3060
Tape marker	-
Landsat header	1440
Geometric transformation record	720
Calibration tables for band 4	1620
" " " " 5	1620
" " " " 6	1620
" " " " 7	1620
" " " " 8	1620
Tape marker	
Data record 1	3780
Data record 9144	3780
Tape marker	-

Table 2

THE DATA TAPES EXAMINED

Tape	RAE No.	Satellite date	Scene	Process date	Type
A	002299	9 July 1977	208/18	5 Apr 1979	Raw
B	002272	9 July 1977	208/18	16 Sept 1979	New cor.
C	001785	2 July 1977	219/24	9 Sept 1979	Old cor.
D	000183	2 June 1976	218/24	2 Jan 1978	Old cor.
E	001855	6 Nov 1975	218/24	30 Oct 1979	New cor.

Table 3

ITEMS IN JSC HEADER RECORD

Item	Byte Nos.	Type	Contents
1	2738-2745	EBCDIC	Sun elevation, milliradians
2	2746-2753	EBCDIC	Sun azimuth, milliradians
3	2885-2886	Binary	Earth rotation, milliradians
4	2887-2894	EBCDIC	Satellite altitude, metres

REFERENCES

No.	Author	Title, etc
1	National Aeronautics and Space Administration	Landsat Data Users Handbook Document No.76SDS4258 (1976)
2	European Space Agency	Format specification for Landsat MSS system corrected computer compatible tape (1976)
3	European Space Agency	Format specification for Landsat MSS system corrected computer compatible tapes produced at Fucino (Italy) (1979)

Overall security classification of this page

UNCLASSIFIED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TR 80018	N/A	UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aircraft Establishment, Farnborough, Hants, UK

5a. Sponsoring Agency's Code	6a. Sponsoring Agency (Contract Authority) Name and Location
N/A	N/A

7. Title A study of Landsat multi-spectral scanner data tapes

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Benny, A.H.			February 1980	36	3

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		Space 574

15. Distribution statement

(a) Controlled by – Head of Space Department, RAE (RAL)

(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

LANDSAT. Multi-spectral scanner. Computer compatible tapes. Corrected.

17. Abstract

A detailed study is described of a number of computer compatible magnetic tapes containing Landsat multi-spectral scanner (MSS) data. This study reveals the differences between raw and corrected data tapes and indicates how the conversion is performed. It also shows the differences between the existing and a new form of corrected data tape.

From the knowledge gained, a computer program has been written, to enable raw data, desirable for certain computer analyses, to be obtained from the existing stocks of corrected data tapes.

RAE Form A143

ILMED

8